INGV-DPC V4 Project

Project V4: "Conception, verification and application of innovative techniques to study active volcanoes"

by

the **Istituto Nazionale di Geofisica e Vulcanologia (INGV)**

and

the **Italian Department for Civil Defense (DPC)**

U.R. V4/10 - WP3.2: *Design and test of a prototype, sea-bottom multi-parametric station integrated to an on-load existing monitoring network.*

Report

A multiparametric low power digitizer: project and results

Massimo Orazi, Rosario Peluso, Antonio Caputo, Marco Capello, Ciro Buonocunto, Marcello Martini

Istituto Nazionale di Geofisica e Vulcanologia

Sezione di Napoli

Osservatorio Vesuviano

October 23, 2007

Contents

1 Introduction.

In the past decade systems and techniques for geophysics monitoring activities have rapidly improved. Monitoring itself is moving to a multi-matter scientific field. The systems under studies often require the collection and analysis of several geophysical parameters. This is, for example, the case of volcano monitoring: volcanoes produces many kind of signals as seismic and acoustic waves, variations in gas composition, deformation of the structure etc. Another example is given by Ocean Bottom Systems (OBS) used as multiparametric geophysics systems: they may be equipped with seismic sensors, hydrophones, electromagnetometers, pressure sensors etc.

The acquisition of all these data (both for experimental and monitoring activities) are usually performed using systems developed by commercial companies. These systems are often conceived only for a single specific application so they may have characteristics not fully satisfying the requirements of such a multiparametric data collection. A typical seismic data recorder provide a number of channels in multiples of three since it is supposed that

the system should acquire data only from tri-axial sensors (velocimeters, accelerometers, etc.). Even the sampling rate of such systems may be only set in ranges useful to acquire seismic data. A multiparametric system should satisfy requirements for different sensors (tilt, seismic, magnetic, etc.) needing different acquisition specifications.

Volcano monitoring often needs instruments installation in remote places, where the only power source is the energy provided by solar panels stored in accumulators. In this case it is important to have electronic instruments requiring very little power consumption in order to minimize the number of solar panels and accumulators to be transported and installed. In the same way low power consumption is an absolute requirement in the case of OBS systems.

Moreover commercial systems are often conceived as closed "black boxes" for which the final user has only limited possibilities to adapt the hardware to his particular needs. For instance the necessity to add some channel in a seismic system is often only possible buying and deploying a new complete system. In some installations is necessary to locate the Global Positioning System (GPS) receiver far from the sensors site (for example an OBS with a moored buoy link or a sensor in a cave) and this may only be done by the manufacturer.

In this context a project named "GILDA" (Geophysical Instrument for Low power Data Acquisition) led by the Italian Institute of Geophysics and Volcanology (INGV), Naples section "Osservatorio Vesuviano" (OV), has been promoted to develop a new data acquisition system that will be able to give an effective answer to these issues for both multiparametric volcanic and OBS monitoring.

The targets of this system are a very low power consumption, modularity and flexibility in order to be adaptable to the specific requirements for different kinds of data acquisition. Good sampling resolutions with an high dynamic range and the possibility to monitor system status parameter (for example the environment temperature) have been took into account. With this project, the INGV also follows the strategic long-term target to autonomously develop the technology necessary for future instrumental applications.

2 System architecture and project.

The first development step in the GILDA project has been the realization of the basic digitizer system presented in this report. It has been conceived as a collection of small subsystems each one dedicated to a specific task following

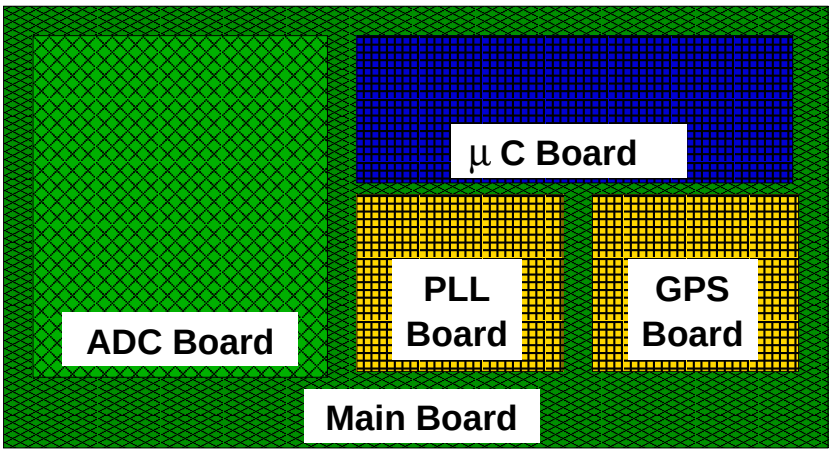


Figure 1: Block schematic of GILDA system.

the general requirements presented in section 1. The project of the digitizer has been greatly simplified by this approach, making the development of the basic system and the testing of the single subpieces a simpler work. This approach results in a little higher production costs offering on the other hand the advantage to improve or change a single sub-board without redesigning everything. The basic system, as schematized in figure 1, is composed by an ADC board with 24 bit resolution, a microcontroller board, a GPS and a PLL board all mounted on a "mother board".

2.1 The ADC board.

The ADC board is based on the CS3301, CS5372 and CS5376A chipsets produced by CIRRUS LOGIC. CS5372 is a dual channel fourth order, high dynamic range, $\Delta\Sigma$ modulator designed for geophysical and sonar applications. The CS5376A is a chip which realizes the DSP functions (essentially a filtering) on the data stream produced by the modulator. When used together they provide an high resolution and an high dynamic range ADC system (nominally 24bits with a bandwidth of $124dB@411Hz$). They offer a low total harmonic distortion, low power consumption per channel and an unique flexibility in geophysical applications so resulting in a valid choice for an ADC board.

The CS3301 is a low-noise, differential input, differential output Programmable Gain Amplifier (PGA). The gain settings are binary weighted ($\times 1$, $\times 2$, $\times 4$, $\times 8$, $\times 16$, $\times 32$, $\times 64$) and may be selected by pin settings. These devices provide a low power running mode which reduces the channel

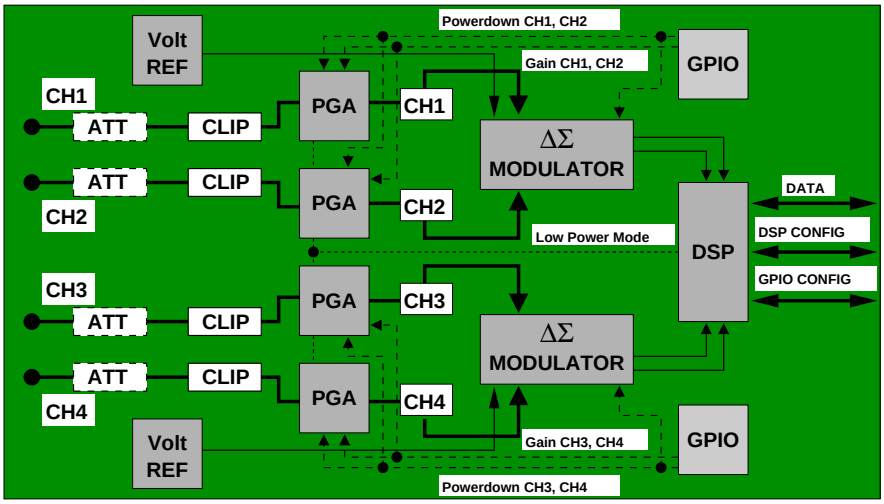


Figure 2: Block schematic of the ADC board.

power requirements lowering, in the meanwhile, the dynamic range of about $3dB$. They also provide an individual power down pin to switch off a single channel. The ADC was realized with a 4 channels configuration in order to preserve the symmetry in the Printed Circuit Board (PCB), when needed one or more of these channels may be turned off using the power down mode. The realized ADC board uses a pair of General Purpose Input Output device (GPIO) used to set up the configuration sent by the microcontroller via the SPI bus. A functional block schematic of the ADC board is shown in figure 2 while in figure 3 there is a picture of an ADC board.

The signal levels of the sensors connected to the board must be adapted to the maximum values tolerated by the ADC chip so it was necessary to introduce in the signal path an input attenuator. A passive resistor attenuator was implemented to avoid a decrease in noise performances due to active circuits. The full scale attenuation ratio is settled by the impedance values of a pair of resistors. A signal line protector and a clipper was placed after the input attenuator to protect the PGA from the destructive effects of overvoltages or transients.

Several low noise design procedures was strongly taken into account to reach the maximum performances of the high resolution ADC. In particular a 4 layer PCB was conceived reserving one of them entirely for the ground plane, one for the power lines and the other two for the signals. The ground plane was partially split into and analog and a digital signals domain connecting them only in one single point under the bottom of the mixed signals

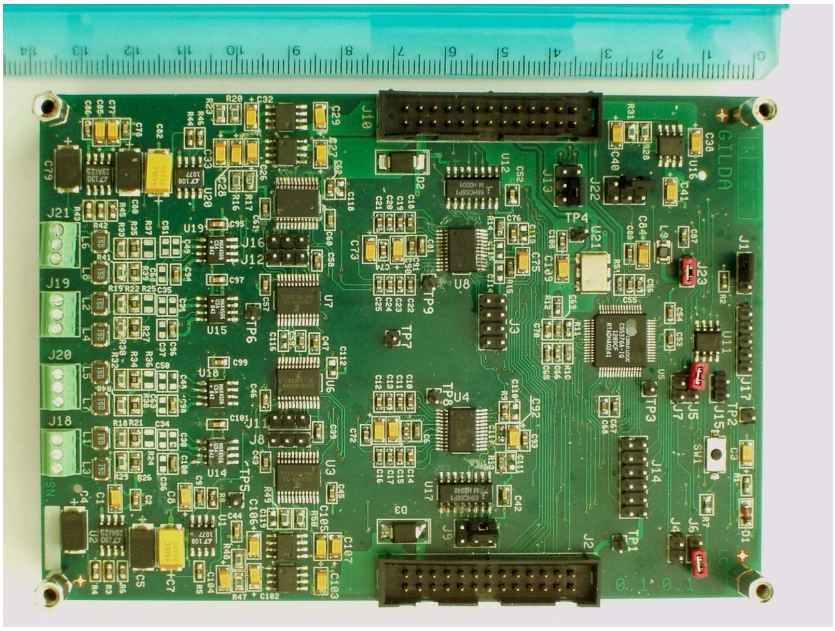


Figure 3: ADC board picture.

components (modulators and PGA). This allows an accurate and separate routing between analog and digital signals avoiding any coupling noise from digital into analog "world".

It is not possible to avoid analog and digital signal proximity in the bottom of the mixed signals components: it may happen that a small amount of digital noise is injected into the analog ground plane. These currents, however, should be quite small since those components do not drive a large fanout. The minimization of fanout on the converter's digital port will also keep its logic transitions relatively free from ringing and it will minimize digital switching currents. This reduces any potential coupling into the analog port of the component. As an extreme caution an optional RC net was reserved in the electrical design of the ADC. This circuit allows cleaning of the digital signals as it works as a shape formator.

The used routing rule is well reported in figure 4 which is a PCB CAD view of the board layers. In brown it is highlighted the ground plane and in black the splitting areas. Red and green shows the signal lines on the top and bottom layers. In the center of the image is possible to view a group of digital (red) lines reaching the target components without crossing the analog ground domain. The separation in the central area of the board prevents the usage of an unique voltage reference for the modulators requiring

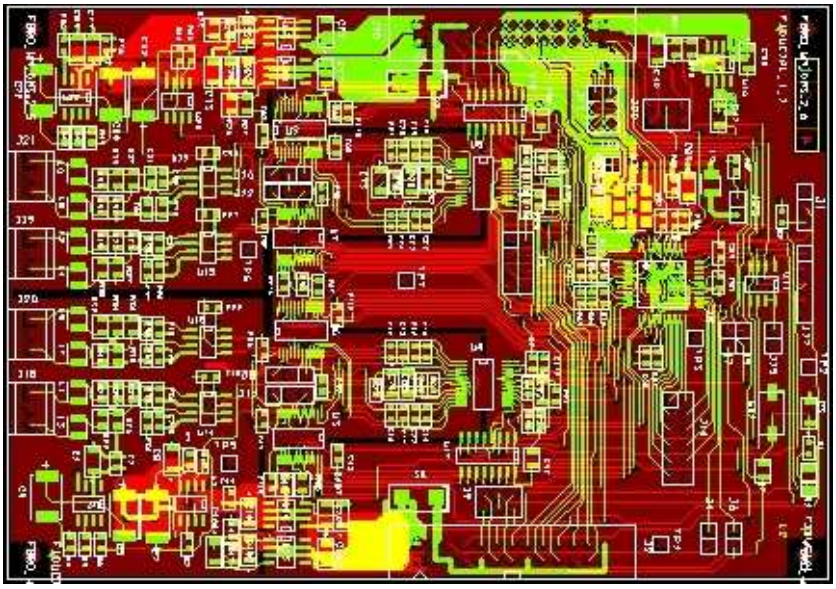


Figure 4: PCB CAD view of the ADC board.

two separate references for each of the two pairs of channels.

The ADC board communicates with the external world via a pair of 26 poles connectors. Data are taken from the DSP via a dedicated serial bus. When solicited on a given line (SDTKI) the DSP will communicate the availability of ready data that must be read as soon as possible. Its internal buffer, in fact, is an 8-deep data FIFO: to avoid any data lost the rate of the SDTKI signal must be equal or greater than the word data rate of the ADC. It is also required to guarantee that the rate of data reading will be high enough.

2.2 The PLL board.

The correct time labeling of each data sample can be reached correlating the CLK clock to the GMT absolute time. This task is accomplished by the PLL board (figures 5 and 6(a)) which operates in a digital controlled phase loop. The external microcontroller modifies the output voltage of the 16 bit DAC changing in this way the frequency of the voltage controlled oscillator. The output clock at $32.768MHz$ is scaled by three onboard counters generating five different synchronous clock speeds. 2, 4, 8 or $16kHz$ clocks may be selectively (via a jumper) sent to the ADC board as the SDTKI signal (see 2.1). The $1Hz$ internal Pulse Per Second (PPS) is sent back to the micro-

controller and is matched with the PPS generated by the GPS receiver. The microcontroller uses a Proportional-Derivative (PD) algorithm (see section 3.1.2) to modify the DAC value and thereby the frequency of the oscillator. This loopback algorithm is able to keep the CLK clock synchronized with the GMT time guaranteeing the correct time labeling of data samples.

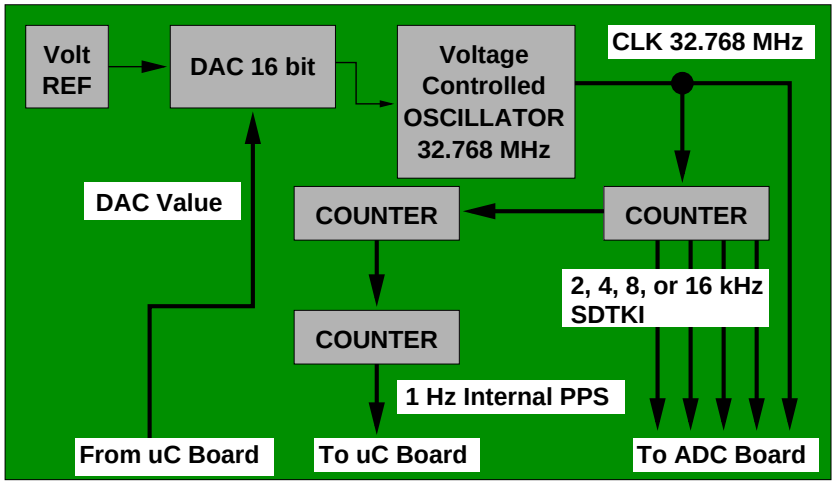
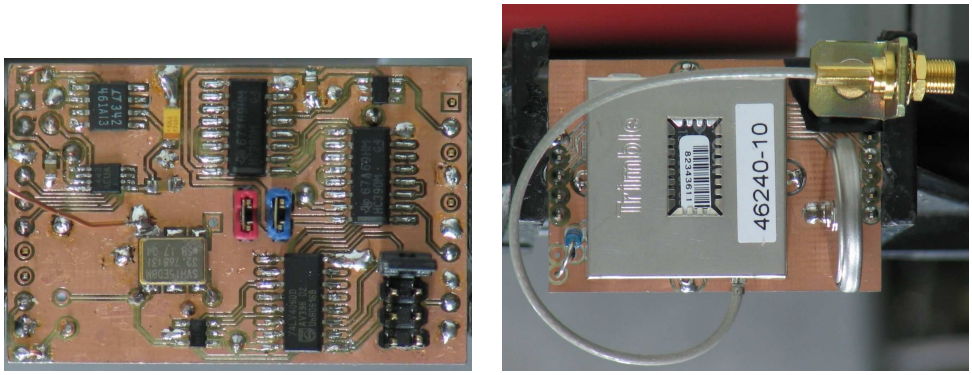


Figure 5: Block schematic of the PLL board.

It is possible to periodically toggle the power of the GPS receiver further reducing the power requirements of the board. This needs some sort of stability in the clock of the $1Hz$ internal PPS. If the oscillator frequency is not an exact submultiple of the PPS (and thus of GMT time) the system adds a constant error in time determination of the samples. Without any correction this error accumulates itself and should not exceed a maximum tolerated value. When the PLL loop is turned off there are many sources of this kind of errors. Changes in the external temperature may make the oscillator drift its frequency, influencing in the same time the voltage reference of the DAC. Fluctuations in the power source may have similiar effects. Another source arises from the discretization of the analog values produced by the DAC: their number directly depends on its bit resolution. Together with the tuning sensibility of the oscillator this determines the minimum frequency difference.

2.3 The GPS board.

The GPS board shown in figure 6(b) is no more than a socket adapter for the complete GPS module. The Trimble Lassen Iq has been chosen



(a) PLL board. (b) GPS board.

Figure 6: Auxiliary boards pictures.

for the GILDA system due to its compact dimensions and very low power requirements. The module receives a set of commands from the microcontroller via its UART interface and sends back informations about time and satellites locking status. It also generates the PPS impulse which is used to synchronize the CLK clock (see section 2.2).

2.4 The microcontroller board.

The microcontroller board is equipped with a 16/32 bits ARM core LPC2129 which accomplish all the supervisor and control logic functions of the system. It synchronizes the GPS PPS with the internal PPS, reads the status of the GPS board via its first UART, reads the data from the ADC board via a dedicated SPI bus and outputs them as a stream on its second UART.

This 32 bit microcontroller has been chosen in order to reduce the consumption of the system. The dynamic power consumption in a logic device is tied to the supply voltage by the relation:

$$P = \frac{1}{2} n C V^2 f$$

that represents the power needed to toggle n logic gates switching once per cycle at a frequency f. C is the capacitance of the single CMOS gate powered at voltage V (note that there is a square dependence on the power voltage). The LPC2129 uses a power core voltage of $1.8V$ against the typical $3.3V$ used by other 8 and 16 bit microcontroller architectures. It follows that a power save factor greater than 3 can be achieved on equal computational

logic resources.

The LPC2129 may work at five different core frequencies (12, 24, 36, 48 and $60MHz$) driven by an internal PLL circuit. Higher clock frequencies are required to reach the highest baudrates of the UART ports. It has two power saving modes realized by shutting down the computational core alone or together with any internal devices. It may be awaken from these idle states by any kind of interrupt requests. The second shutdown mode also turns off the internal counters used to measure lags between events and it has too long awakening times. For these reasons it is unuseful for our kind of application. Using just the first shutdown mode allows the system to save a nice amount of power as the microcontroller spends many time idling in this state. As the core is completely shut down, the power consumption of the microcontroller becomes quite independent from the controller clock.

The firmware running on the controller and some of the used algorithm are briefly described in section 3.

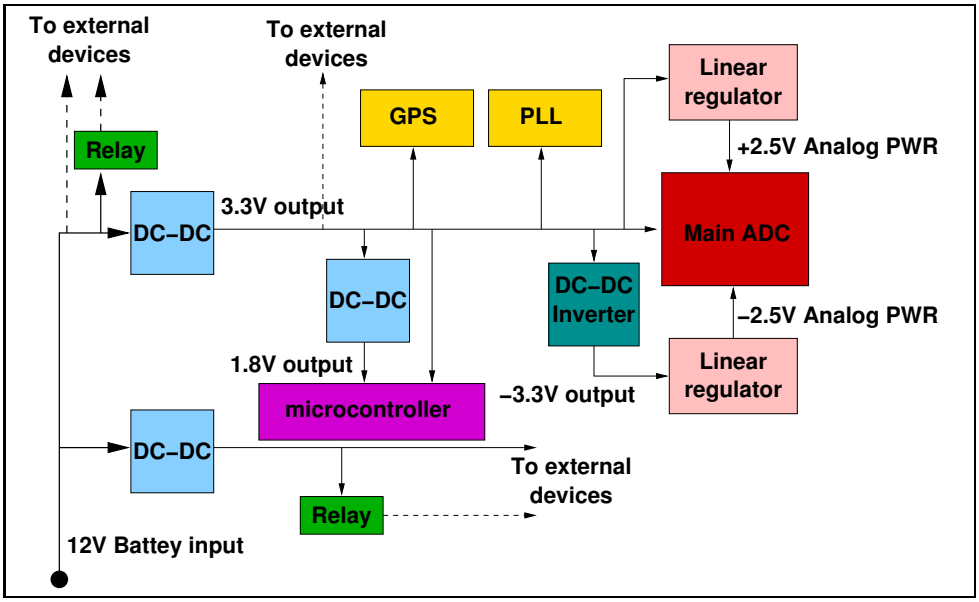


Figure 7: Schematic block of the power net of the GILDA system.

2.5 The main connection board.

The main board of the GILDA system interconnectes all the other boards previously described. It contains the main power source section that feeds all the other ones (see figure 7). DC-DC converters have been used to drop

down battery voltage from the input value ($9 \div 18V$) to 3.3 and $5V$. The $3.3V$ power line serves all the logic devices with the exception of the microcontroller core which requires $1.8V$ generated by another DC-DC. Two low-noise power rails are needed for the mixed signals section of the main ADC board. A low noise dropout linear regulator gives the $+2.5V$ from the $3.3V$ regulated power while a charge pump voltage inverter first produces the $-3.3V$ which is then raised to the required $-2.5V$. With the exception of these last lines all the other ones are directly accessible by reserved connectors on the main board. Some of them are under relay control in order to toggle the power of externally connected devices.

A $12bit$ ADC is also present on the main board. It is used to obtain device status informations, such as temperature, power consumption, battery voltage, solar panel current or any other auxiliary low-rate acquisition. This ADC can in fact used at a maximum rate of 1 sample per second. It has single ended inputs with maximum range selectable at $12V$, $5V$ or $3.3V$.

In addition the board has an expansion serial port to connect other general purpose cards for future applications. One will probably be a middle-resolution $16bit$ ADC for low-rate data acquisition.

3 Firmware description.

The system firmware is written in standard Ansi C using the GNU GCC ARM toolchain (version 3.3.1 using uClib 0.9.20). Some small GCC extensions to the Ansi C were required to handle interrupt functions. The use of ARM machine language has been kept to the possible minimum: only few lines were used to enable/disable interrupt requests. The uLink JTAG (Joint Test Action Group) interface is currently used to download and debug the program into the internal microcontroller Flash memory. This method will not be used in production boards as the LPC2129 allows its programming through the first UART interface.

Care has been taken during the development to keep the firmware divided into big functional blocks. In this way both the development and the debugging of the code was simpler and faster. Adding new features to the code is also simplified by a well factorized code. In figure 8 there is a rough block description of the firmware tasks and of their interconnections.

The board has three main different modalities of operation. Eeach of them performs specific tasks and has been written having in mind the maximum ease of use and the minimum possible power consumption. This has been achieved, for example, turning off any unused device, or avoiding idle loops on the microcontroller making it to enter shutdown mode when possible.

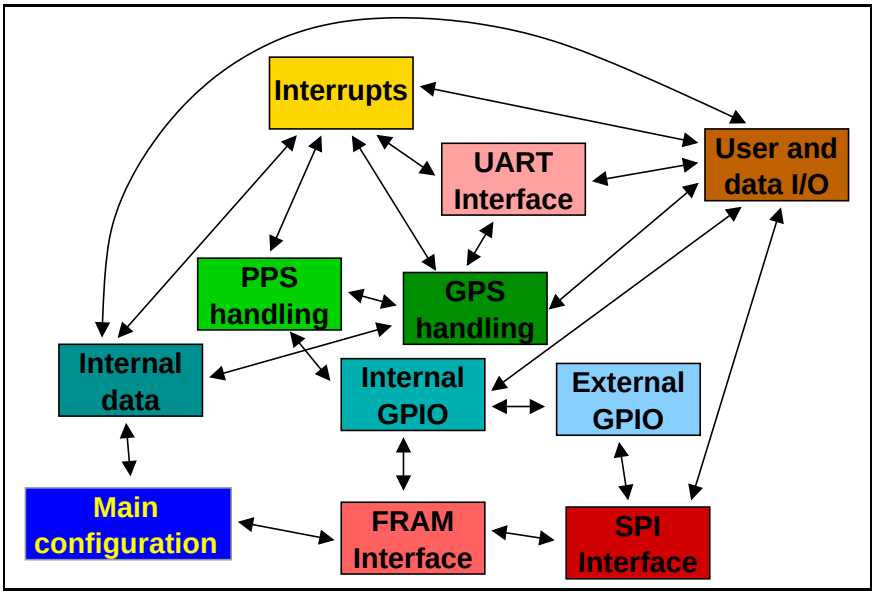


Figure 8: Logical blocks of the GILDA firmware.

Currently active modes are:

1. The boot sequence.
2. The acquisition loop.
3. The configuration and control menu.

The main interface of the firmware is the second UART port of the LPC2129 (being the first one reserved to GPS communications). By default the station starts at 57600*baud* but it may be configured to work at any known standard serial speed. Non standard speeds (230400*baud* and 460800*baud*) are currently supported for devices that may use them, while the 921600*baud* speed is not supported by the current hardware. No special program is needed to configure and handle the station: a simple serial terminal emulator on the serial line is pretty good.

During the boot sequence or the acquisition loop the station will answer to simple commands sent on the serial line. For example three `'c'` characters in a row (ASCII value `0x63` hexadecimal) will make the station switch to the configuration menu. Other commands may be sent, a complete list is reported in table 1.

Commands	**ASCII equivalents** (hexadecimal)	**Actions**
`ccc`	`0x63 0x63 0x63`	Enter configuration mode
`rrr`	`0x72 0x72 0x72`	Perform a soft reset
`hhh`	`0x68 0x68 0x68`	Perform an hard reset[a]
`bbb`	`0x62 0x62 0x62`	Enter binary configuration mode[b]

[a]Hard reset is only available using an external circuit. When the extra board is not present it will revert to a soft reset.

[b]Binary configuration is not yet implemented in the firmware, but the board already answers to this command.

Table 1: Command strings available during the boot sequence and the acquisition loop. Each command must be sent within one second.

3.1 The boot sequence.

The program enters this state when turned on or after a reset. This phase is devoted to the initialization and configuration of the board hardware. It performs many tasks, for example the measurement of the internal microcontroller clock and the first phase lock between the clock generated by the PLL board (see section. 2.2) and the PPS impulse coming from the GPS receiver.

Figure 9 shows a schematic flux diagram of the boot procedure. The only interrupt routine active during the procedure is the handler of user inputs on the second UART interface. It parses the input characters checking for one of the commands of table 1. When a command is recognized it halts the boot sequence and enters the requested firmware state.

3.1.1 GPS configuration.

Communications with the GPS receiver go through the first UART interface using the binary Trimble Standard Interface Protocol (TSIP). This has been used instead of the standard NMEA protocol because it contains packets for both configuring the receiver and acquiring GPS navigation data.

The receiver configuration is compared with the one it should have for a correct work. Only if it mismatches (for example for a newly mounted receiver) it is sent again and stored in the GPS internal Flash memory.

After checking (or sending) the configuration, the microcontroller waits for the first fix of the GPS position with more than three used satellites. Only when correctly locked, in fact, the PPS signal from the receiver is

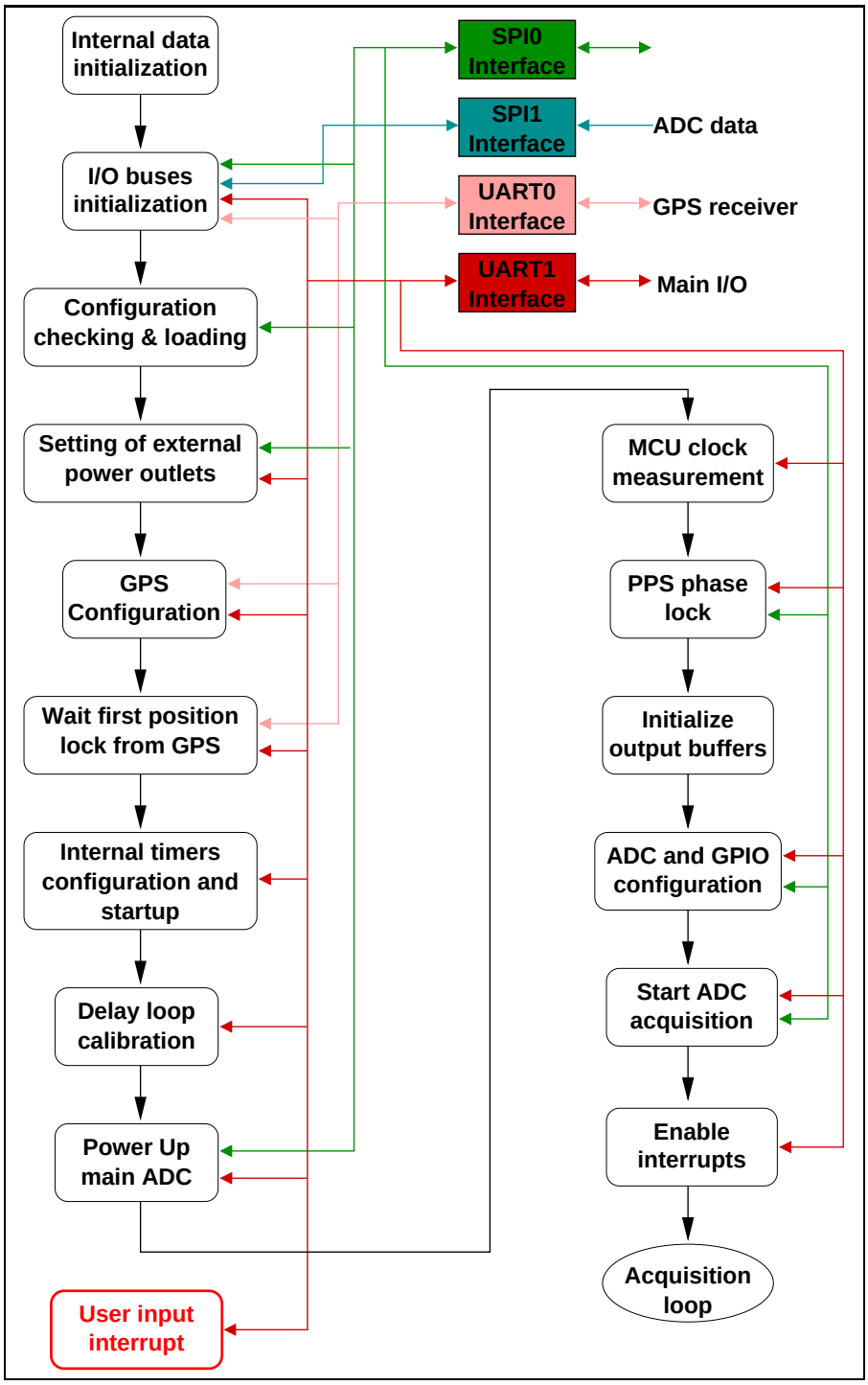


Figure 9: Schematic flux diagram of the boot sequence.

guaranteed to be stabilly related to the absolute GMT time. The MCU use these pulses also to make a more precise measure of its internal clock. This is required to correct the group delay introduced by the ADC in the data samples (see section 3.2).

3.1.2 PPS locking.

The first phase lock between the CLK clock and the PPS sent by the GPS is performed during the board boot. It may last for many minutes (up to ten) in order to reach the required locking precision needed by the board. The lock may be done between the external PPS and the SDTKI clock or between the external PPS and the one generated by the PLL board. This can be selected by configuration options and by some switches on the mother board of the station.

An internal timer of the MCU runs freely incrementing at each clock of the core. Each rising edge on the PPS's (or on the external PPS and the SDTKI signal) triggers a snapshot of the counter value. The difference between two consecutive snapshots gives a measure of the phase difference of the two signals (with the resolution given by the clock).

The PPS locking loop uses a Proportional-Derivative (PD) algorithm to keep this phase as little as possible. The measured phase difference is used to calculate the next value sent to the 16 bit DAC using the following recursive equation:

$$D_i = D_{i-1} + \alpha P_i + \beta(P_i - P_{i-1}),$$

where D_i are the values sent to the DAC, P_i are the measured phases, α and β two experimentally determined constants. The lock is considered good when the measured phase remains under a given threshold for 60 consecutive times. The best values for the two constants have been experimentally determined as those that make the locking procedure as fast as possible while remaining stable. Using the actual hardware they are:

$$\begin{cases} \alpha = -0.5 \\ \beta = -2 \end{cases}$$

The same algorithm is used during the following acquisition loop where phase measurement and DAC setting are made into dedicated interrupt routines. In figure 10 is reported a schema of the currently used PLL loop.

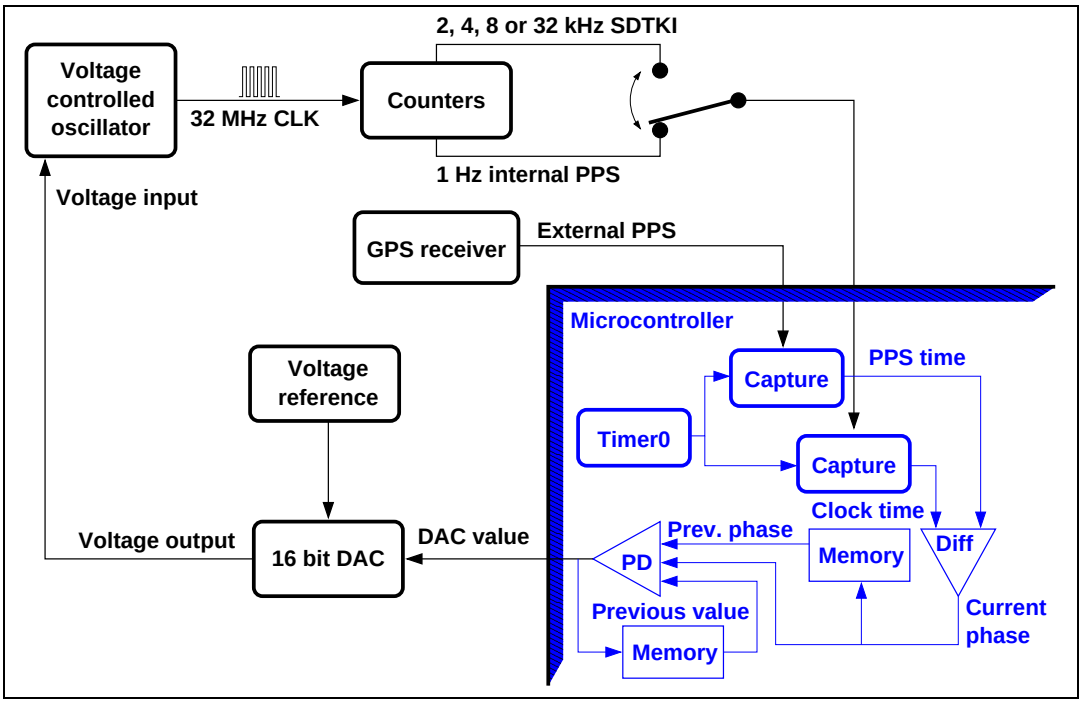


Figure 10: PLL loop schema.

3.1.3 Boot ending.

When the PPS signal is locked the firmware sends the configuration to the ADC board via the SPI bus and then the start command. Interrupts are temporarily disabled, the interrupt vector is cleared and the new interrupt mask for the acquisition loop is loaded. Then the boot procedure exits and the acquisition loop is entered.

3.2 The acquisition loop.

This is the main operational mode of the digitizer system. A schematic flux diagram is shown in figure 11. Here the data are read from the ADC, packed and sent on the serial line. Most of the operations are done via interrupt requests (IRQ). The main loop of the firmware just waits for new data to be available and sends them as soon as they are ready. The microcontroller is then put in its first power saving mode (see section 2.4). Each interrupt request awakes the core and a new loop is done. The firmware never exits from this loop until turned off, reset or when sent in the configuration menu continuing to output the data stream at a rate of one packet per second.

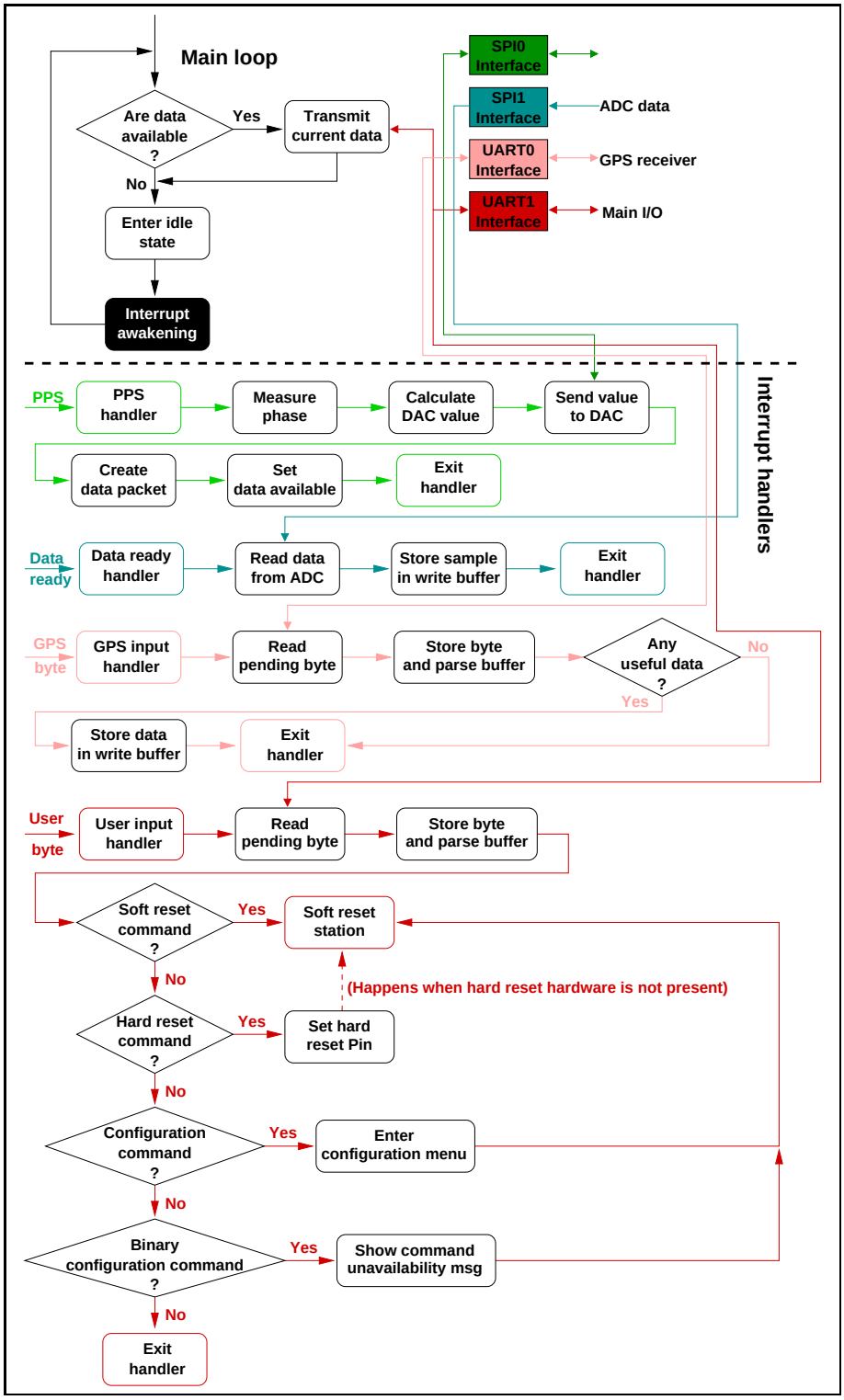


Figure 11: Schematic flux diagram of the acquisition loop.

The most important IRQ active in this phase is the one answering to the PPS's. The board may be configured to react both to the GPS impulse or to the PLL board impulse. In this second case (which is the default one) the firmware can also perform temporary shutdowns of the GPS receiver.

The first PPS interrupt is used to send the SYNC signal to the ADC. This signal resets all its internal buffers and forces the acquisition to restart. The signal is sent some time before the next PPS in order to correct the ADC group phase delay. Subsequent interrupts are used to correct the CLK clock frequency in the same way described in the 3.1.2 section. Also, using the informations got from the GPS receiver, the time of the last packet is calculated and the "data available" flag is set for the main loop.

The ADC, stimulated by the SDTKI signal, sends a data ready signal (SDRDY) each time it has data to be read. This signal is connected to an external interrupt request of the microcontroller. The interrupt routine extracts data via the second dedicated SPI interface. They are then packed in a write buffer in the controller RAM.

The last two interrupts are active on the two serial lines. The one on the first UART reads and parses the data coming from the GPS receiver setting all the needed flags and data in the currently written packet. The second one is active on the other UART interface and answers to user inputs as it does during the boot procedure.

3.3 Board configuration.

When the board enters configuration mode, all the external devices are shut down. Only the GPS receiver, the main SPI bus and the two UART lines are kept on. Configuration is done through easy to use number driven menu. In figure 12 is shown a screenshot of the main configuration menu. Note that the menus described in this article are related to the current version of the firmware, namely $0.12.x$.

Briefly, the item **'1'** will show the current board configuration as eventually modified. Item **'2'** allows to modify many board parameters: a new menu, shown in figure 13, is then entered. Changes do not become effective until item **'6'** is used which then exits from the configuration menu. Item **'3'** is used to revert any change done and come back to the previous board configuration. Item **'7'** discards the modification done and exits from the configuration menu. Item **'5'** is used to turn on and off the auxiliary power outputs ($5V$ or $12V$) acting on some relais present on the mother board (into another menu not shown there).

All the configuration parameters of the board are contained into a Ferromagnetic RAM (FRAM). Item **'8'** is used to format this FRAM and to

```
************************************************************
GILDA-Controller v.0.12.0 "Lilith"
************************************************************

Main configuration menu:
         1 - Show current configuration.
         2 - Modify configuration.
         3 - Restore previous configuration.
         4 - GPS module handling.
         5 - Configure power outputs.
         6 - Save configuration and exit.
         7 - Discard configuration and exit.
         8 - Factory reset.
         9 - Perform a soft reset.
        10 - Perform an hard reset.

Select:
```

Figure 12: Screenshot of the main configuration menu.

write into it the default ("factory") configuration. Item **'9'** resets the microcontroller and restarts the firmware from the boot sequence. Item **'10'** will only work if the external hard reset circuit is mounted. In this case the power is temporarily shut down to all the board. When not present the microcontroller will "manually" shut down the power to the external devices and then perform a soft reset.

Item **'4'** enters another menu (figure 14) where you can handle the GPS receiver. Here it is possible to check and send the configuration of the receiver, look for its current lock status, turn off and on the receiver or make it perform one of the supported resets.

4 Conclusions and results.

The basic digitizer system is shown in the photo in figure 15 (page 22) where is possible to see all the boards described in section 2.

4.1 Power consumption.

Power consumption of the realized digitizer depends on many configurable settings: active channels, DSP filter clock and power modality. Even the configurable $5V$ power output (see 2.5) drains an additional current when

```
**********************************************************************
GILDA-Controller v.0.12.0 "Lilith"
**********************************************************************

Alter configuration:
1. Name of station: GLD0                2. Samples per second: 100
3. Number of channels: 4                4. Serial baud rate: 57600
        1 - CH01        2 - CH02        3 - CH03        4 - CH04

5. Filter clock frequency: 16384 kHz    6. Master clock frequency: 2048 kHz
7. MDATA clock frequency:  512 kHz      8. MCU frequency: 12 MHz
9. Output get at: FIR2 output           10. Low power mode: DISABLED

11. GPS mode: ENHANCED  12. Wait first GPS fix: YES (Ignored after: 120)

13. Data transmission: BINARY           14. Boot mode: NORMAL
15. Lock mode: PPS                      16. PPS Source: PLL board

17. Auxiliary ADC: ENABLED (using 2 channels)
Channels: 1 - AUX0/1 (1) 2 - AUX1/2 (2)

18. Previous menu.

Select:
```

Figure 13: Screenshot of the modification menu.

```
**********************************************************************
GILDA-Controller v.0.12.0 "Lilith"
**********************************************************************
GPS handling menu (38400 baud):
         1 - Identify GPS serial speed.
         2 - Retrieve and check GPS configuration.
         3 - Acquire GPS statuses.
         4 - Send configuration to GPS.
         5 - Store configuration to GPS flash.
         6 - Power cycle GPS unit.
         7 - Soft reset GPS.
         8 - Hard reset GPS.
         9 - Factory reset GPS.
        10 - Return to previous menu.

Select:
```

Figure 14: Screenshot of the GPS control menu.

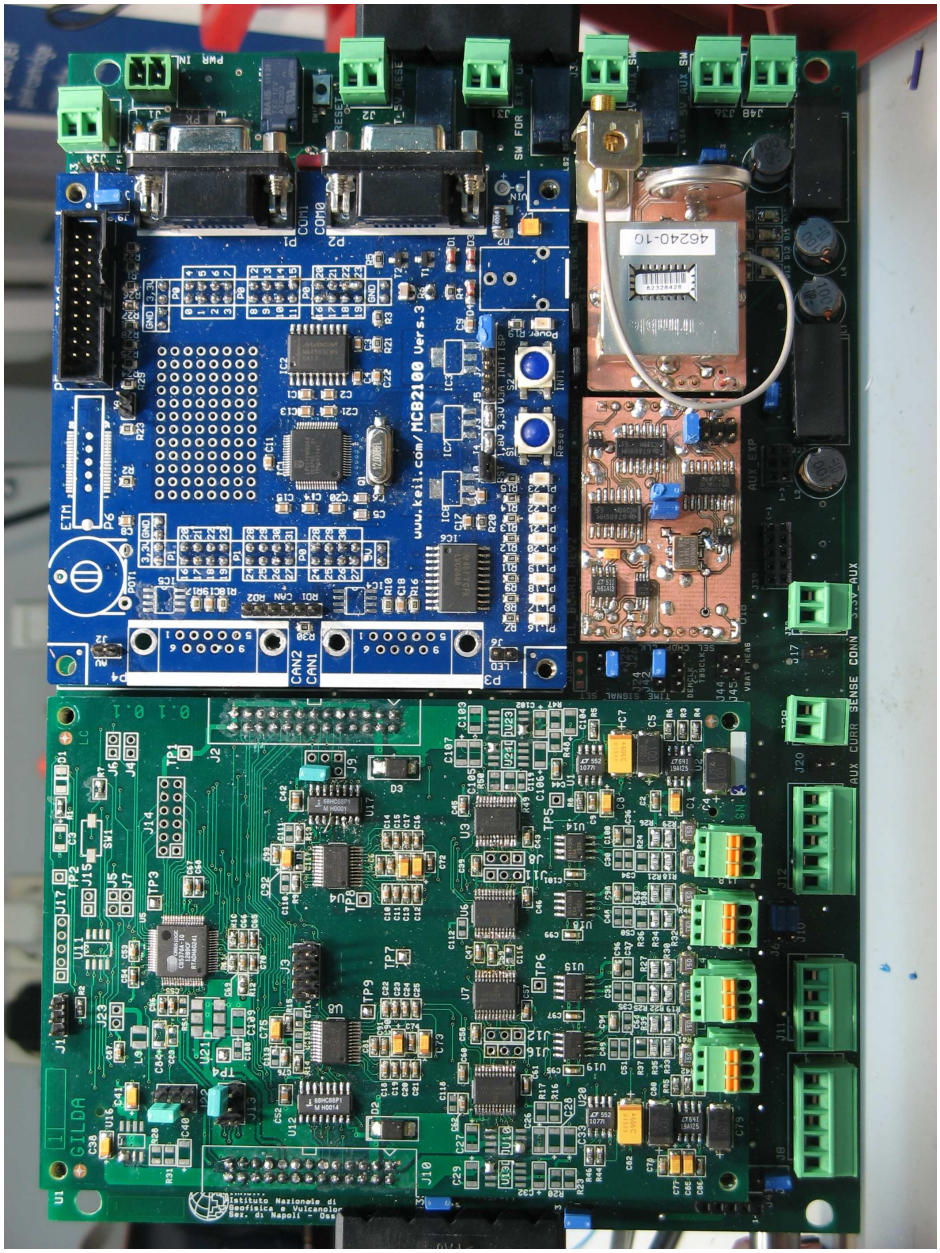


Figure 15: Photo of the first GILDA digitizer prototype.

turned on, raising the total consumption of the board. When this line is turned off the total absorbed current lies in the range from $\simeq 44.55mA$ to $\simeq 70mA$ (@$12V$) depending on the configuration. A complete list of the measured power consumptions is in table 2.

Low power mode	OFF				ON			
Active channels	1	2	3	4	1	2	3	4
DSP clocks	Power consumptions (mW)							
$16MHz$	624	696	768	840	600	642	678	720
$8MHz$	588	660	732	804	564	600	642	684
$4MHz$	570	642	708	786	540	582	624	666
$2MHz$	558	630	696	768	534	576	618	660

Table 2: Summary of power consumptions for the GILDA digitizer.

The used CS5376A, however, does not support any DSP clock at any output word rate. There are some limitations on the minimum power consumption at those rates due to the minimum value of the clock. The supported DSP clock frequencies are shown in table 3.

4.2 Noise characteristics.

Noise characteristics and dynamic range of the ADC system (DSP + modulator) are directly related to the oversampling ratio. The modulator oversamples the signal at $512kHz$, then the DSP decimates and filters it giving the requested Output Word Rate (OWR). Increasing the OWR makes the oversampling ratio ($512kHz$ / OWR) to decrease and, consequently, the noise level to increase.

	DSP clocks (MHz)			
Word rates	2.048	4.096	8.192	16.384
$40Hz$	NA	NA	NA	OK
$50Hz$	OK	OK	OK	OK
$100Hz$	NA	NA	OK	OK
$125Hz$	OK	OK	OK	OK
$200Hz$	NA	NA	NA	OK

Table 3: Supported DSP clock rates.

This can be view comparing figures 16 (page 26), 17 (page 27) and 18 (page 28). They represent three noise acquisitions of the same ADC channel shorted on a 50Ω[1] resistor at three different sampling rates ($50Hz$, $100Hz$ and $200Hz$ respectively) in the non low-power mode. In the figures 19 (page 29), 20 (page 30) and 21 (page 31) are shown the same acquisitions made in low-power mode. They have been done using the maximum DSP clock frequency and using the idle tones removal pin setting of the modulator. This introduces a fixed offset on the signal that has been removed in the figures. Noise appears to be gaussian distributed around its mean (random noise). The width of the gaussian distributions increases as the output word rate increases. The Signal-to-Noise Ratio (SNR), expressed in dB, is given by:

$$\mathrm{SNR}_{dB} = 20\log_{10}\left(\frac{C_{\mathrm{FS}\ rms}}{C_{\mathrm{NOISE}\ rms}}\right)$$

where $C_{\mathrm{FS}\ rms}$ is the rms of the full scale input, while $C_{\mathrm{NOISE}\ rms}$ is the rms of the noise both expressed in digitizer counts. SNR measurement are usually performed using a fixed frequency sine wave as input signal. In such case, the rms is given by:

$$C_{\mathrm{FS\ sine}\ rms} = \frac{C_{pp}}{2\sqrt{2}}$$

being C_{pp} the peak-to-peak value of the signal. As reported by the data sheets of the modulator a full scale differential input signal will be translated in the interval $[6291455, -6291455]$. This will give to a theorical full scale sine wave input signal an rms of:

$$C_{FS\ rms} = \frac{6291455}{\sqrt{2}} \simeq 4448730$$

expressed in digitizer counts. Table 4 summarizes the measured rms noise and the SNR_{db} of the digitizer.

The ADC system has been conceived to be interfaced to different sensors with different output voltages changing the input attenuator value (see section 2.1). So it is not possible to give a fixed relation between input voltage and output counts. This can be only done using the maximum modulator input, which is fixed by its voltage reference in the interval $[-2.5V, 2.5V]$. The single count conversion value is then:

$$V_{\mathrm{count}} = \frac{2.5V}{6291455} \simeq 397.4nV$$

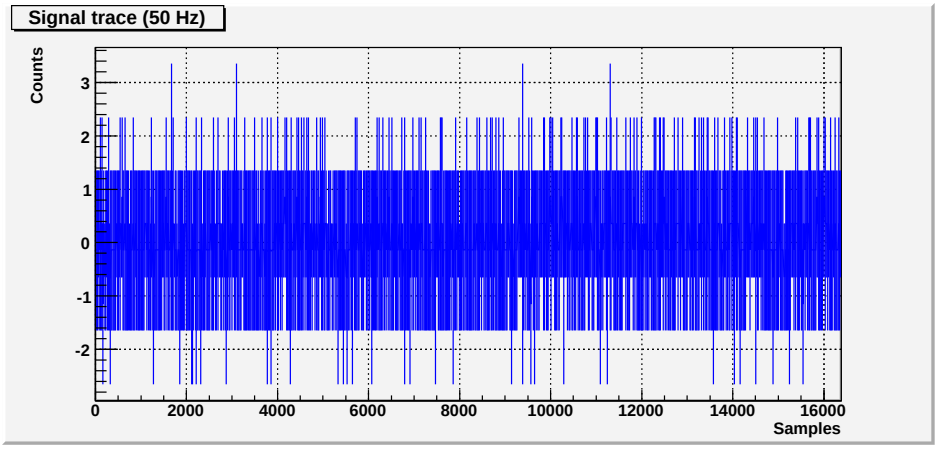
[1]This is needed to simulate the typical output impedance of an active sensor.

Acquisition	$C_{\text{NOISE } rms}$	SNR_{dB}
$50Hz$ normal	0.8	134.9
$50Hz$ low power	0.9	133.8
$100Hz$ normal	1.0	132.9
$100Hz$ low power	1.2	131.4
$200Hz$ normal	1.5	129.4
$200Hz$ low power	1.6	128.9

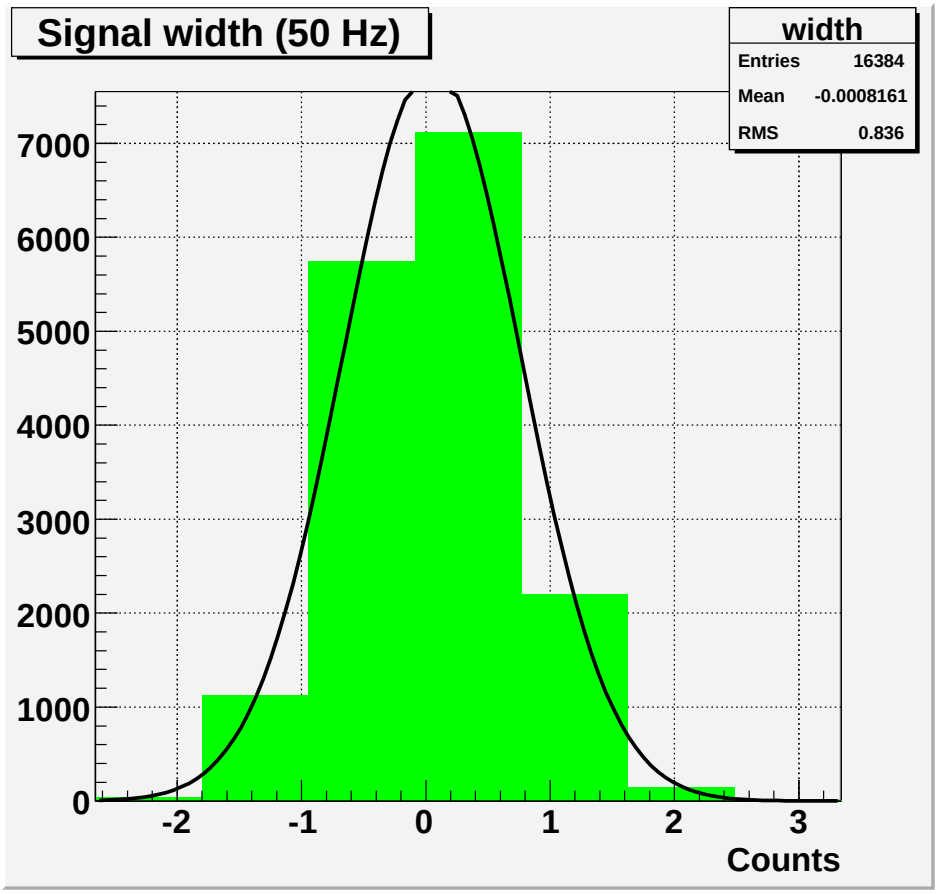
Table 4: *rms* noise (shorted input) and relative SNR_{db} at several output word rates.

that can be reported to the effective maximum input signal multiplying it by the input attenuator factor. For example, the ADC used in the reported tests has a $4\times$ attenuator, resulting in an accepted input in the interval $[-10V, 10V]$ with a voltage per counts factor of:

$$V_{\text{count } 4\times} = 4 \times V_{\text{count}} \simeq 1.590\mu V.$$

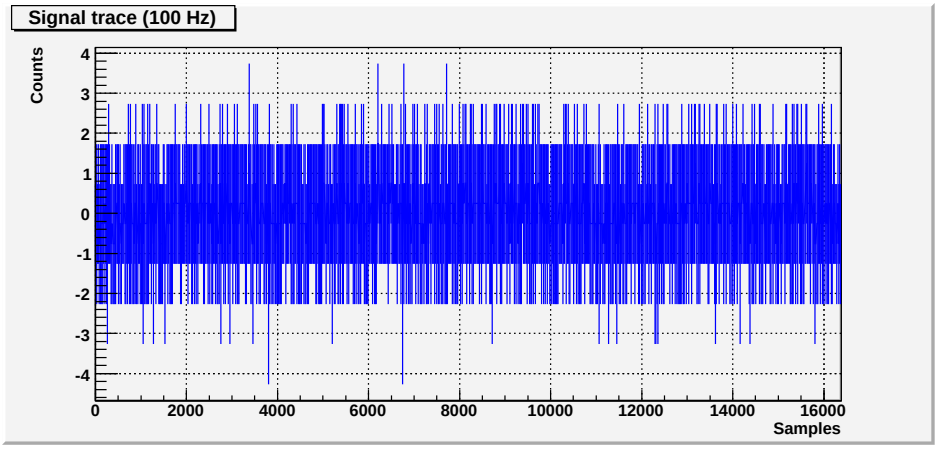


(a) Trace.

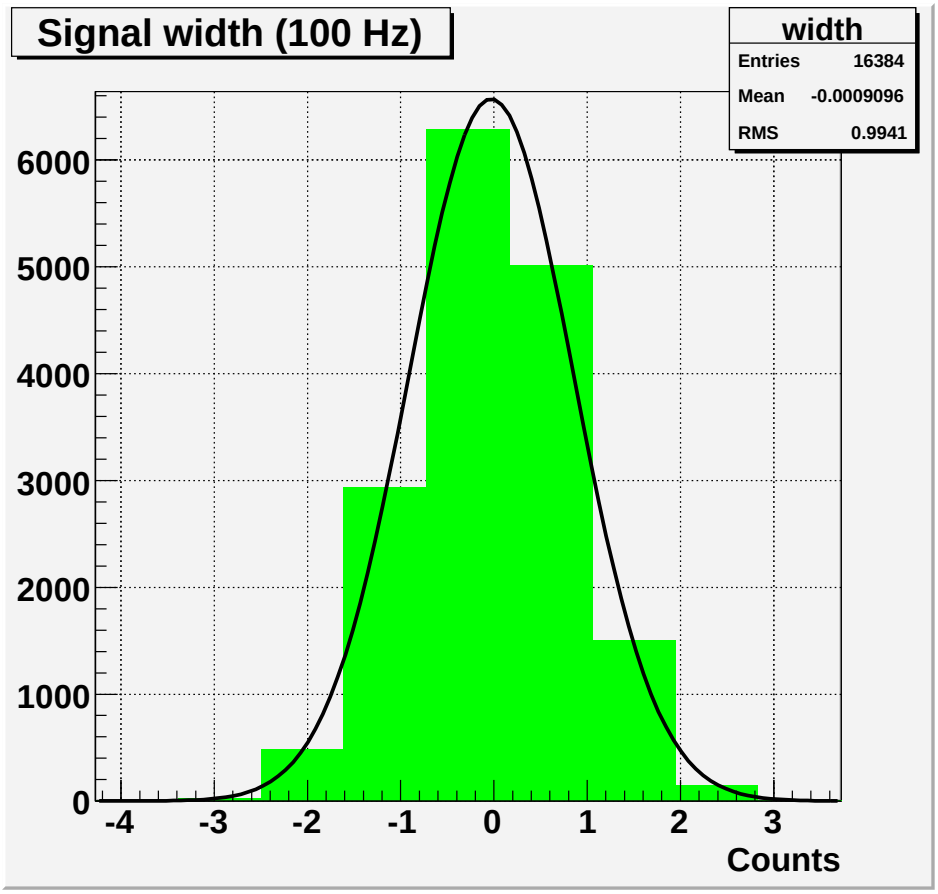


(b) Signal histogram.

Figure 16: Shorted input noise recorded at $50Hz$.

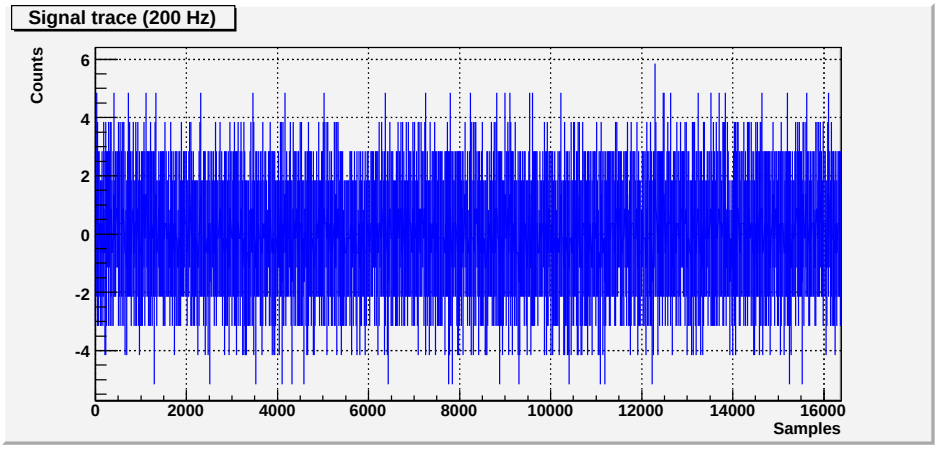


(a) Trace.

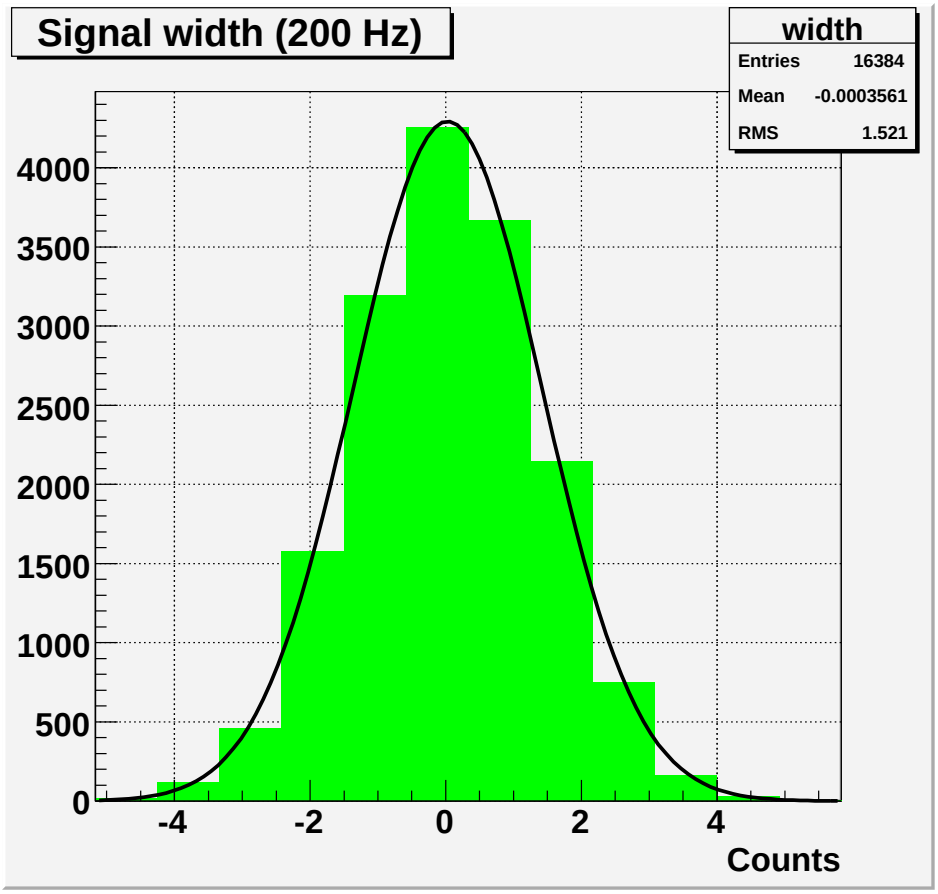


(b) Signal histogram.

Figure 17: Shorted input noise recorded at $100Hz$.

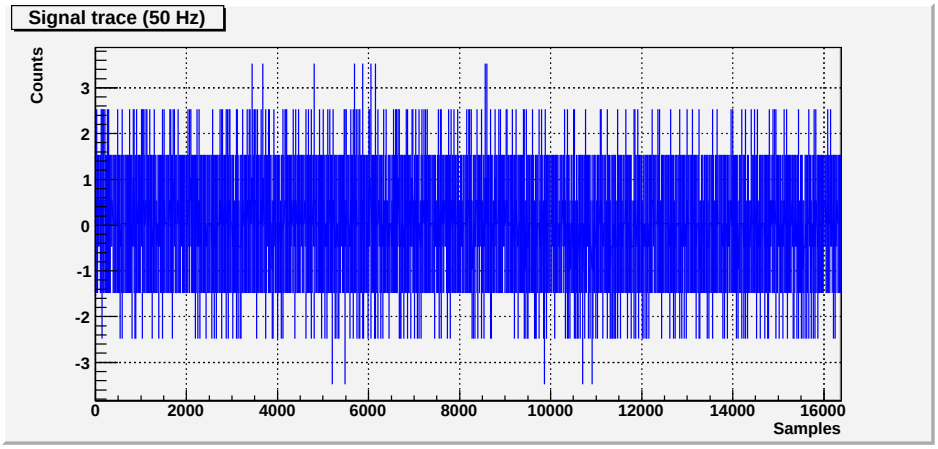


(a) Trace.

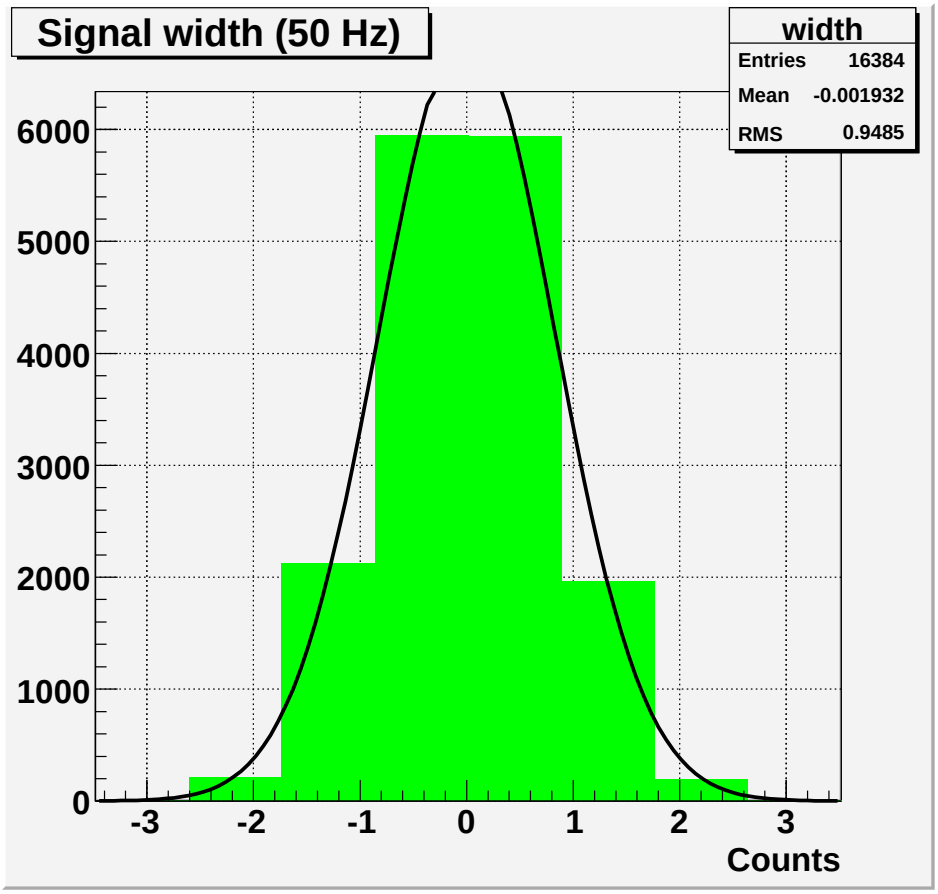


(b) Signal histogram.

Figure 18: Shorted input noise recorded at $200Hz$.

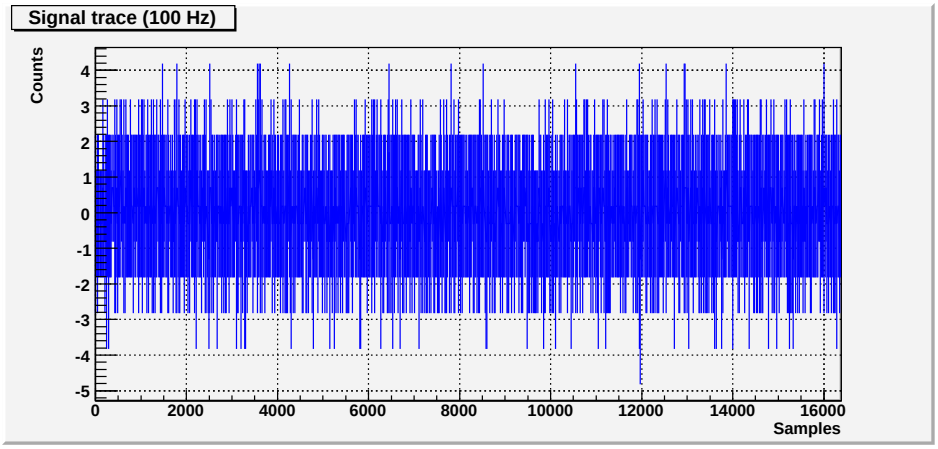


(a) Trace.

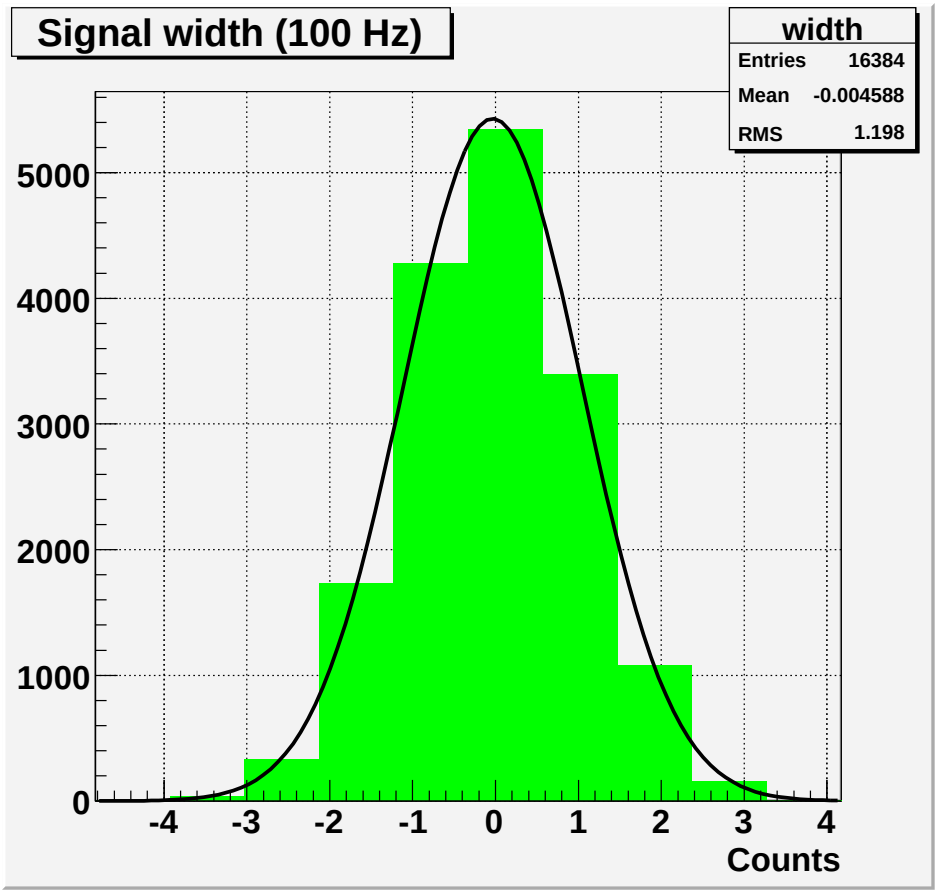


(b) Signal histogram.

Figure 19: Shorted input noise recorded at $50Hz$ in low power mode.

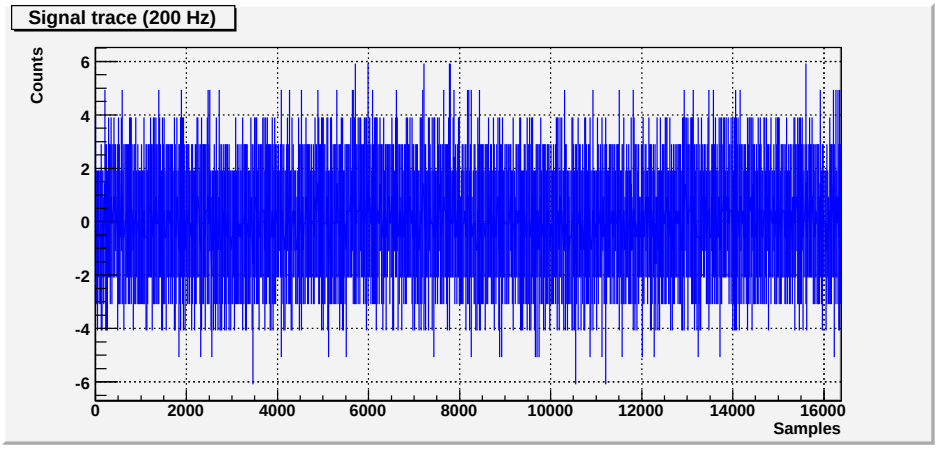


(a) Trace.

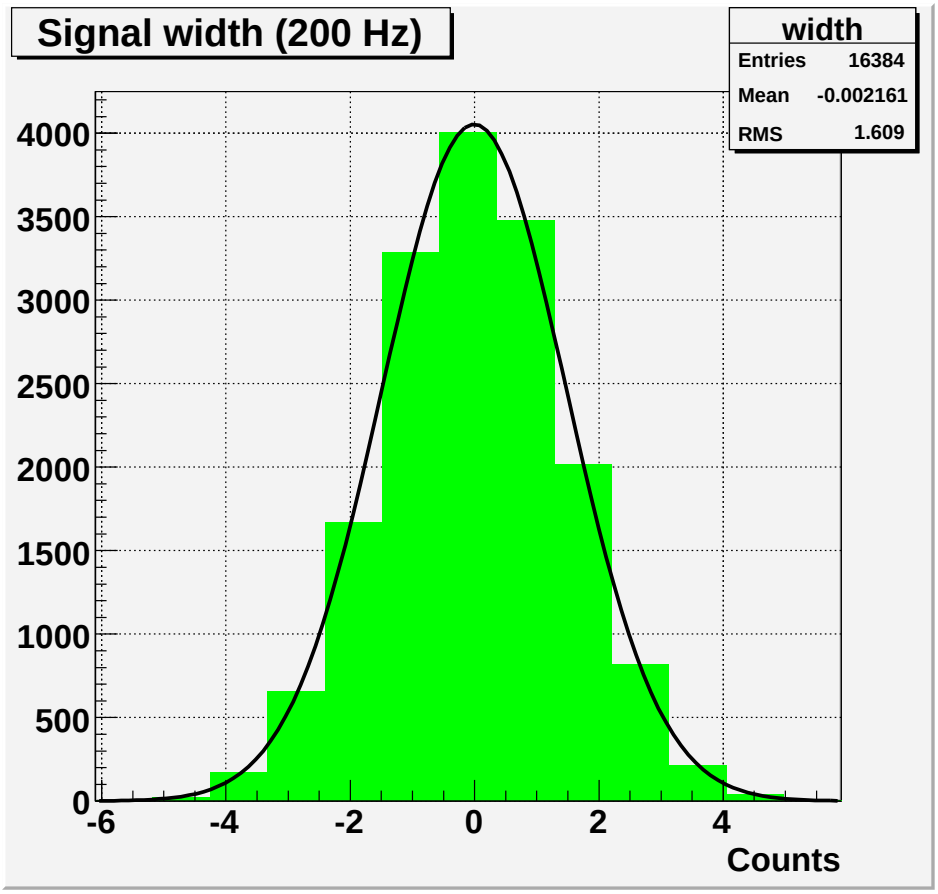


(b) Signal histogram.

Figure 20: Shorted input noise recorded at $100Hz$ in low power mode.



(a) Trace.



(b) Signal histogram.

Figure 21: Shorted input noise recorded at $200Hz$ in low power mode.